

REMARKS/ARGUMENTS

The present amendment is submitted in accordance with the Revised Amendment Format.

The Examiner has objected to claims 4 and 11 of this Application due to informalities.

The Examiner has rejected claims 1-6, 12-17, and 19 of this Application under 35 U.S.C. § 103(a) as being unpatentable over “Getting Started with SNAP™,” by Template Software, Inc. (hereinafter “START”), in view of “Using the SNAP™ Development Environment,” by Template Software, Inc. (hereinafter “SNAPDEV”) under.

The Examiner has rejected claims 7 and 18 of this Application under 35 U.S.C. § 103(a) as being unpatentable over START, as modified by SNAPDEV, and further in view of “Using the SNAP™ Language,” by Template Software, Inc. (hereinafter “LANG”).

The Examiner has rejected claim 8 of this Application under 35 U.S.C. § 103(a) as being unpatentable over “Using the SNAP™ External Application Software Component,” by Template Software, Inc. (hereinafter “EXT”), in view of START.

The Examiner has rejected claim 10 of this Application under 35 U.S.C. § 103(a) as being unpatentable over START in view of U.S. Patent 6,748,455 B1 to Hinson et al. (hereinafter “Hinson”).

The Examiner has rejected claim 11 of this Application under 35 U.S.C. § 103(a) as being unpatentable over “Using the SNAP™ Communication Component,” by Template Software, Inc. (hereinafter “COM”), in view of U.S. Patent 5,426,747 A1 to Weinreb et al. (hereinafter “Weinreb”).

Claims 4 and 11 have been amended.

All amendments are fully supported by the specification and no new matter has been added.

Reconsideration and allowance in view of the amendments and remarks is respectfully requested.

Claim Objections

The Examiner has objected to claims 4 and 11. Further, the Examiner requested that in claim 4, line 3, “the component” should read “the reusable component,” and in claim 11, line 7, “interface context data and configuration context data” should read “interface data and configuration data.” Applicant has amended claims 4 and 11 with the appropriate corrections as requested by the Examiner.

Applicants respectfully submit that this objection be withdrawn.

Rejection under 35 U.S.C. § 103(a) based on START and SNAPDEV

The first issue in this case is whether claims 1-6, 12-17, and 19 are unpatentable over START in view of SNAPDEV under 35 U.S.C. § 103(a). The Examiner asserts that START teaches some of the limitations of claim 1. Applicants respectfully disagree with the Examiner. Claim 1 recites:

“A computer program product, tangibly embodied in a machine-readable storage device comprising instructions operable to cause data processing apparatus to:
 implement a reusable software component encapsulating functionality,
 multiple instances of the component being usable at the same time;
 the reusable component having at least one visual representation;
 the reusable component having a programming interface for programmatic interaction with the reusable component;
 the reusable component having a data-binding interface for data communication with the reusable component; and
 the reusable component having a visual interface for access to the at least one visual representation of the reusable component.”

(Claim 1) (Emphasis added).

The Examiner cited the Object Model component in Figure 5-1 on page 5-3 of START. The figure teaches an object model containing “SNAP Language,” an “Inference Engine” and an “Event Handler.” Clearly, the Examiner’s citation does not teach to “implement a reusable software component encapsulating functionality, multiple instances of the component being usable at the same time.” Neither “SNAP Language,” nor an “Inference Engine” nor an “Event

Handler” disclose or even suggest the limitation of “multiple instances of the component being usable at the same time.”

With respect to the other limitations of claim 1 except for the “the reusable component having at least one visual representation” limitation, the Examiner relies on the same Figure 5-1 in alleging that START teaches those limitations. The text preceding and referring to Figure 5-1 states:

“Figure 5-1 shows a high-level view of the predefined software provided by the five components of the Foundation Template.”

(START, pg. 5-3) (Emphasis added).

In addition, the caption for Figure 5-1 provides:

“The predefined software provided by the Foundation Template components”

(START, pg. 5-3, Figure 5-1) (Emphasis added).

It is evident that Figure 5-1 teaches software having multiple components, demonstrated by START’s description of Figure 5-1 having “five components” and its reference to the boxes within Figure 5-1 as “Foundation Template components.” Claim 1, on the other hand, does not disclose multiple components. Instead, claim 1 only discloses a single “reusable component” that contains all the limitations expressed in claim 1. This is clearly supported by the fact that all the limitations in claim 1 contain “the reusable component” language, referring to the same reusable component. In other words, each limitation that the Examiner alleged START teaches is contained in a separate and distinct component whereas each limitation of claim 1 is associated with in a single reusable component. Having each limitation associated with its own respective component, resulting in the existence of multiple components, is obviously different than associating each limitation with a single component.

Since START does not teach the limitation of “multiple instances of the component being usable at the same time” and each of the limitations that START allegedly teaches is found in separate and distinct components, START does not teach all the limitations

contended by the Examiner in claim 1. Therefore, claim 1 is not unpatentable over START in view of SNAPDEV under 35 U.S.C. § 103(a).

Claims 2-6 are dependent claims of claim 1. Thus, these claims are allowable for at least the same or similar reasons.

Independent claim 12 recites the following limitations:

“A computer implemented method, comprising:
implementing a reusable software component encapsulating functionality,
multiple instances of the component being usable at the same time;
the reusable component having at least one visual representation;
the reusable component having a programming interface for programmatic interaction with the reusable component;
the reusable component having a data-binding interface for data communication with the reusable component;
the reusable component having a visual interface for access to the at least one visual representation of the reusable component; and
storing the reusable component.”

(Claim 12) (Emphasis added).

Claim 12 includes the same limitations as claim 1 and thus is allowable for the same or similar reasons stated for claim 1. Claims 13-17 are dependent claims of claim 12 that include all the limitations of claim 12 and additional limitations. Therefore, these claims are allowable for at least the same or similar reasons

Independent claim 19 provides the following:

“An apparatus, comprising:
means for implementing a reusable software component encapsulating functionality, multiple instances of the component being usable at the same time;
the reusable component having at least one visual representation;
the reusable component having a programming interface for programmatic interaction with the reusable component;
the reusable component having a data-binding interface for data communication with the reusable component; and
the reusable component having a visual interface for access to the at least one visual representation of the reusable component.”

(Claim 19) (Emphasis added).

Claim 19 includes the same limitations as claim 1 and thus is allowable for the same or similar reasons stated for claim 1.

Applicants respectfully submit that this rejection has been overcome.

Rejection under 35 U.S.C. § 103(a) based on START, SNAPDEV, and LANG

The second issue in this case is whether claims 7 and 18 are unpatentable over START, as modified by SNAPDEV, as applied to claims 1 and 12 above, and further in view of LANG under 35 U.S.C. § 103(a). Applicants respectfully disagree.

Claim 7 is a dependent claim of claim 1 and includes all the limitations of claim 1 and additional limitations. It is respectfully submitted that the addition of LANG fails to teach, indicate or suggest the features discussed above regarding claim 1. Thus, for the same or similar reasons stated for claim 1, claim 7 is allowable. Claim 18 is a dependent claim of claim 12 and includes all the limitations of claim 12 and additional limitations. It is respectfully submitted that the addition of LANG fails to teach, indicate or suggest the features discussed above regarding claim 12. Accordingly, claim 18 is allowable for the same or similar reasons stated for claim 12.

Applicants respectfully submit that this rejection has been overcome.

Rejection under 35 U.S.C. § 103(a) based on EXT, START

The third issue in this case is whether claim 8 is unpatentable over EXT in view of START under 35 U.S.C. § 103(a). Applicants contend otherwise. Claim 8 provides:

“A computer program product, tangibly embodied in a machine-readable storage device, the computer program product comprising instructions operable to cause data processing apparatus to:

implement an application runtime framework, the framework being operable to:

receive a specification of a component interface to be used in an application without a specification of a corresponding component implementation, the component interface having a programming interface, a data-binding interface, and a visual interface; and

instantiate a particular component implementation at application

runtime, the particular component implementation being selected from one or more component implementations corresponding to the component interface”

(Claim 8) (Emphasis added).

The Examiner claims that EXT teaches the limitation to “receive a specification of a component interface to be used in an application without a specification of a corresponding component implementation.” The Examiner refers to a “handle” disclosed in EXT as support. EXT states “a handle points, or refers, to an element or a sequence of elements in a class definition, or to a parsed value.” Certainly, a handle that “points, or refers, to an element or a sequence of elements in a class definition, or to a parsed value” is not the same as to “receive a specification of a component interface to be used in an application.” Hence, EXT does not teach the “receive a specification of a component interface to be used in an application” part of the limitation.

Furthermore, since EXT does not teach the limitation “to “receive a specification of a component interface to be used in an application,” it also cannot teach to “receive a specification of a component interface to be used in an application without a specification of a corresponding component implementation.” Nonetheless, the citation referred by Examiner does not mention, nor does EXT disclose, the limitation that the “specification of a component interface” is “to be used in an application without a specification of a corresponding component implementation.” Instead, the citation used by the Examiner merely states “[u]sing handles is more efficient than using strings to access SNAP elements” and thus does not disclose the particular limitation “to be used in an application without a specification of a corresponding component implementation.”

Because EXT does not teach the limitation to “receive a specification of a component interface to be used in an application without a specification of a corresponding component implementation,” claim 8 is not unpatentable over EXT in view of START under 35 U.S.C. § 103(a) and is thus allowable.

Applicants respectfully submit that this rejection has been overcome.

Rejection under 35 U.S.C. § 103(a) based on START and Hinson

The fourth issue in this case is whether claim 10 is unpatentable over START in view of Hinson under 35 U.S.C. § 103(a). Applicants do not agree with the Examiner's characterization of the cited prior art. Claim 10 recites the following:

“A computer program product, tangibly embodied in a machine-readable storage device, for implementing an application runtime framework, the computer program comprising instructions operable to cause data processing apparatus to: receive an event subscription directed to a subscribing component when the subscribing component has not been initiated, the event subscription specifying subscriptions to one or more events generated by sub-components embedded by the subscribing component;
cache events generated by the sub-components that are specified by the event subscription while the subscribing component has not been instantiated; and
forward any cached events to an instance of the subscribing component after the subscribing component is instantiated.”

(Claim 10) (Emphasis added).

For the limitation “receive an event subscription directed to a subscribing component when the subscribing component has not been initiated,” the Examiner refers to this portion of Hinson for support:

“In the case of a persistent subscription which specifies the subscribed by class identifier, the firing control **202** creates the subscriber if not yet instantiated”

(Hinson, col. 15, lines 40-43) (Emphasis added).

Moreover, Hinson provides:

“The firing control **202** controls delivering the event for a particular subscription.”

(Hinson, co. 15, lines 36-37) (Emphasis added).

Hinson teaches “delivering the event for a particular subscription.” Applicant's limitation teaches “receiv[ing] an event subscription.” Evidently, “delivering” is dissimilar, and in fact opposite, to “receiving.” Notwithstanding such differences, Hinson only teaches “creat[ing] the subscriber if not yet instantiated,” whereas Applicant's limitation qualifies the “receives an event subscription” with the limitation “when the subscribing component has not been initiated.” That

is, the “receives an event subscription” limitation occurs only “when the subscribing component has not been initiated.” Therefore, Hinson does not teach the limitation “receive an event subscription directed to a subscribing component when the subscribing component has not been initiated.” Also, START does not teach this limitation nor does the Examiner claim that it does. Indeed, the Examiner has expressly stated in the rejection that START does not disclose this limitation.

The Examiner rejected Applicant’s limitation “cache events generated by the sub-components that are specified by the event subscription while the subscribing component has not been instantiated” by citing the following:

“The subscription cache **204** temporarily stores the subscriptions per each method of the event class object’s outgoing-event interface for use by the event dispatcher **200**, so as to allow more rapid distribution of events to the subscribers **106-108**.”

(Hinson, col. 15, lines 58-62).

So, Hinson teaches “[t]he subscription cache [] temporarily stores the subscriptions per each method of the event class object’s outgoing-event interface for use by the event dispatcher.” Applicant’s limitation discloses “cache events generated by the sub-components that are specified by the event subscription while the subscribing component has not been instantiated.” In other words, Hinson’s teaching that “[t]he subscription cache [] temporarily stores the subscriptions per each method of the event class object’s outgoing-event interface” is conditioned on “for use by the event dispatcher.” On the other hand, Applicant’s limitation discloses “cache events generated by the sub-components that are specified by the event subscription” conditioned on “while the subscribing component has not been instantiated.” Obviously, the condition “for use by the event dispatches” is different than the condition “while the subscribing component has not been instantiated.” Therefore, Hinson does not teach Applicant’s limitation “cache events generated by the sub-components that are specified by the event subscription while the subscribing component has not been instantiated.” Likewise, START does not teach this limitation nor does the Examiner claim that it does.

Since neither Hinson nor START teaches the limitations “receive an event subscription directed to a subscribing component when the subscribing component has not been initiated” and “cache events generated by the sub-components that are specified by the event subscription while the subscribing component has not been instantiated,” claim 10 is not unpatentable over START in view of Hinson under 35 U.S.C. § 103(a) and is thus allowable.

Applicants respectfully submit that this rejection has been overcome.

Rejection under 35 U.S.C. § 103(a) based on COM and Weinreb

The fifth issue in this case is whether claim 11 is unpatentable over COM in view of Weinreb under 35 U.S.C. § 103(a). Applicants differ with the Examiner’s interpretation COM’s teachings. Claim 11 states the following:

“A computer program product, tangibly embodied in a machine-readable storage device, for implementing an application runtime framework, the computer program product comprising instructions operable to cause data processing apparatus to:

receive one or more context mappings for a component, the context mappings being specified by a component embedder to exchange context data with the component, the context data comprising interface data and configuration data;

if the component has not been instantiated, cache the specified context mappings; and

create the specified context mappings for the component after the component has been instantiated.”

(Claim 11) (Emphasis added).

The Examiner maintains that COM teaches the limitation “create the specified context mappings for the component after the component has been instantiated.” The Examiner cites to the following:

“These mappings take effect when the connection is made.”

(COM, page 5-4, line 36) (Emphasis added).

Applicant's limitation teaches to "create the specified context mappings," but COM teaches that the "mappings take effect." The limitation to "create the specified mappings" is not the same as the limitation of "mappings take effect." In fact, the limitations are very different. Applicant's limitation specifically focuses on "creat[ing] the specified mappings." It necessarily follows then that the "specified mappings" did not exist before and thus had to be "create[d]." COM's teachings specifically focuses on "mappings tak[ing] effect." It is reasonable that the mappings were already created before "tak[ing] effect." Since COM teaches "mappings take effect," it obviously does not teach the Applicant's particular limitation to "create the specified context mappings."

In addition, COM fails to teach the other part of the limitation to "create the specified context mappings for the component after the component has been instantiated." The Examiner's citation provides that "[t]hese mappings take effect when the connection is made." So, the "mappings take effect" limitation is restricted by "when the connection is made." That is to say, the "mappings take effect" only "when the connection is made." Applicant's limitation to "create the specified context mappings for the component" is restricted to "after the component has been initialized." Since COM only teaches the "mappings take effect when the connection is made," it does not teach the limitation "to "create the specified context mappings for the component after the component has been instantiated." Similarly, Weinreb does not teach this limitation nor does the Examiner claim that it does.

Seeing that neither COM nor Weinreb teach the limitation to "create the specified context mappings for the component after the component has been instantiated," claim 11 is not unpatentable over COM in view of Weinreb under 35 U.S.C. § 103(a) and thus is allowable.

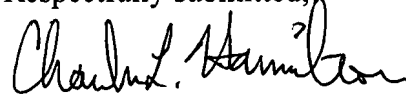
Applicants respectfully submit that this rejection has been overcome.

CONCLUSION

In view of the foregoing, Applicants believe all claims now pending in this Application are in condition for allowance. The issuance of a formal Notice of Allowance at an early date is respectfully requested.

If the Examiner believes a telephone conference would expedite prosecution of this application, please telephone the undersigned at 408-244-6319.

Respectfully submitted,

A handwritten signature in cursive script, appearing to read "Charles L. Hamilton".

Charles L. Hamilton
Reg. No. 42,624

FOUNTAINHEAD LAW GROUP P.C.
900 Lafayette Street, Suite 509
Santa Clara, CA 95050
Tel: 408-244-6319
CLH:LD